

CS 276, FALL 2004

HOMEWORK 1: DUE OCTOBER 8TH, 5:00PM

The objective of this assignment is to familiarize you with NS-2. There are two aspects to NS-2:

1. Using the tools already present in NS-2
2. Changing NS-2 to suit your needs.

In this assignment you are expected to do a bit of both. The first part involves changing the C++ code in NS-2 by writing your own ping agent. You are allowed to follow Marc Gries' tutorial. The goal is to guide you through the tutorial. However, the more you figure out on your own, the better off you will be for later assignments.

Part 1

Declare a class PingAgent which is a subclass of Agent. Define all functions necessary and declare and implement all other classes that you feel are necessary. Then write Tcl code to test the operation of your ping agent.

1. To begin with, consider a simple topology:

N1——N2

There should be one ping agent stationed at each of the two nodes. Node N1 should ping the node N2. N1 should print the RTT whenever it receives a response.

2. Now consider the following topology:

N1——N2——N3——N4——.....——Nm

We want to find out the transfer time for each segment of the path from N1 to Ni. Based on the input parameter (i.e., 2, 3...m), N1 should print onto the screen the transfer time for each segment of the path from N1 to Ni (i.e. from N1 to N2, N2 to N3, ... N(i-1) to Ni)

Modify your code to accomplish this. Though you MAY make changes in C++ to accomplish this, you can also write OtcI instprocs for Agent/PingAgent. Note that for two agents to "talk", you must write \$ns connect \$agent1 \$agent2

Some of the credit for this part of the assignment is for how you design mechanisms in NS to accomplish this. Whatever your design you must point out its advantages or disadvantages in the write-up.

Part 2

In this part you will rewrite the ping agent entirely in OTcl. You will use the Tcl class Agent/Message to accomplish this. Since the ping agent must send ping requests at regular intervals, you must write a small Timer Class as well.

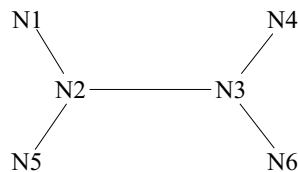
To get you started with this part, you can use the template file myping.tcl. You may use this skeleton to develop your code.

1. Repeat the same exercises as in Part 1
2. —same as in Part 1—

Part 3

In this part you will study how background traffic and lossy links affect your results.

1. Consider the following topology:



Station ping agents at N2 and N3.

Station FTP applications at N1 and N5 talking with their counterparts at N4 and N6, respectively.
Assume full duplex links.
Start all the applications at the same instant of time.

Answer the following questions in your write-up:

1. How does throughput for the TCP transfer vary with the sending rate of the ping agent? Choose the link capacities and other parameters suitably so that you "see" something happening.
2. Is the bandwidth shared "fairly" between the FTP applications? If yes, why? If not, what do you think is happening?
3. If the FTP applications are started at different times, how do the results vary for 1. and 2. ? Explain whatever you observe.

Grading: Your report will be graded based on your explanation of what occurs during the simulations. For each of the problems, you should explain how the various parameters, such as queue size, link capacity, link lossiness etc. impact the results. You should also explain why you chose whatever parameters you chose in running your simulations.

Turn-in: Submit a writeup (maximum five (5) pages) about what you did in this assignment, the design and parameter choices you made and why you made them. In effect, summarize your work. Also be sure to answer all the above questions. In addition, you must also turn in a print out of all of your code. Submit your report by October 8th, 5.00pm in the homework box for CS276, Room 2108 Eng 1.